1. **Superkey and Candidate key:**

In a relational database, a key is a set of one or more attributes that can be used to uniquely identify a record in a table. Superkey and Candidate key are two types of keys used in database design.

A superkey is a set of one or more attributes that can uniquely identify a record in a table. However, a superkey may contain additional attributes that are not required for uniqueness. In other words, a superkey is a superset of a candidate key.

A candidate key, on the other hand, is a minimal superkey, meaning that it is a set of attributes that can uniquely identify a record in a table, and no subset of the key can also uniquely identify the record. In other words, a candidate key is a superkey with no redundant attributes.

For example, consider a table of students, with columns for student ID, name, and email. The combination of student ID and email would be a superkey because it uniquely identifies each student. However, since name is not required for uniqueness, the combination of student ID and email would also be a candidate key, while the combination of all three attributes would be a superkey but not a candidate key.

1. **Primary key and Unique key:**

Primary key and Unique key are two constraints used to enforce the uniqueness of values in a table.

A primary key is a special type of unique key that is used to uniquely identify each record in a table. It must be unique and not null for every record in the table. Additionally, a table can have only one primary key. In most cases, the primary key is defined using a single column, but it can also be defined using multiple columns if necessary.

A unique key, on the other hand, is used to ensure that the values in a column or set of columns are unique. Unlike a primary key, a table can have multiple unique keys. However, unlike a primary key, a unique key does not have an automatic index associated with it.

For example, in a table of customers, the customer ID column might be defined as the primary key, while the email column might be defined as a unique key to ensure that no two customers have the same email address.

**3.Draw and write about all the symbols**

In database design, various symbols are used to represent different elements of the database model. The symbols are used to provide a visual representation of the model, making it easier to understand and communicate.

1. Entity Symbol: The entity symbol is a rectangular box that represents a table in the database. The name of the table is written inside the box. The attributes of the table are written below the name of the table.
2. Attribute Symbol: The attribute symbol is an oval shape that represents an attribute in a table. The name of the attribute is written inside the oval.
3. Primary Key Symbol: The primary key symbol is a double-lined oval that represents the primary key of a table. It is drawn around the attribute that is the primary key.
4. Foreign Key Symbol: The foreign key symbol is a key symbol with a single line that represents the foreign key in a table. It is drawn around the attribute that is a foreign key.
5. Relationship Symbol: The relationship symbol is a diamond shape that represents the relationship between two tables. It is drawn between the two tables that are related. The type of relationship is written inside the diamond.
6. One-to-One Relationship: A one-to-one relationship is represented by a line connecting two tables with a single line. The line connects the primary key of one table to the foreign key of the other table.
7. One-to-Many Relationship: A one-to-many relationship is represented by a line connecting the primary key of one table to the foreign key of another table. The line has a crow's foot symbol on the many side.
8. Many-to-Many Relationship: A many-to-many relationship is represented by a line connecting two tables with crow's foot symbols on both sides. In this case, a junction table is used to represent the relationship.

Overall, these symbols are an essential part of database design as they provide a visual representation of the relationships and structure of a database. By understanding these symbols, one can create and communicate effective database designs.